

Available Fields

namespace where
all form fields
can be found

use Filament\Form\Components

Amount

TextInput

The total amount

Checkbox Published Published

Toggle Is admin Is admin Is admin Is admin

CheckboxList

Planets

Select all

Mercury Venus Earth Mars

Planets

Mercury Earth Venus Mars

Radio

Like this post? Yes No

Like this post? Yes No

Like this post? Yes No

ColorPicker

Color

Textarea

Description

Learn how to use Filament components to write awesome applications

TagsInput

Tags

Tailwind CSS Alpine.js

Select

Status

Domain

MorphToSelect

Addressable

User × ↓

Select an option ↓

KeyValue

Meta

	Key	Value	
↑↓	description	Filament is a collection of Laravel packages	🗑️
↑↓	og:type	website	🗑️
↑↓	og:site_name	Filament	🗑️
Add row			

ToggleButtons

Status

Draft

Scheduled

Published

Like this post?

Yes No

Technologies

Tailwind CSS

Alpine.js

Laravel

Laravel Livewire

DateTimePicker

Published at

dd /mm /yyyy , -- :-- :-- 📅

At

▶ 00:00:00 🕒

Date of birth

dd /mm /yyyy 📅

Date

Starts 01/01/2000 📅 at midnight

Markdown

Content

B *I* ~~S~~ 🔗 **H** “ ” </> ☰ ☷ 📄 🖼️ ↺ ↻

RichEditor

Content

B *I* ~~S~~ 🔗 **H** Heading Subheading “ ” </> ☰ ☷ 📄 🖼️ ↺ ↻

Upload

Attachment

Drag & Drop your files or [Browse](#)

ColorPicker Field

use `Filament\Forms\Components\ColorPicker;`

pick a color in a range of formats

```
ColorPicker::make('color')
```

Methods

```
->hex()
->hsl()
->rgb()
->rgba()
```

◦ set the color format. Default: hex()

Color

#FF80C0

Textarea Field

use `Filament\Forms\Components\Textarea;`

interact with a multi-line string

```
Textarea::make('description')
```

Description

Learn how to use Filament components to write awesome applications

Methods

```
->autosize() auto resize to fit the content
->rows(10)   textarea rows
->cols(20)   textarea columns
->readOnly() make textarea read only
```

Validation specific to Textarea

```
->minLength(2)
->maxLength(1024)
->length(100)
```

TagsInput Field

also supports [spatie/laravel-tags](#)

use `Filament\Forms\Components\TagsInput;`

interact with a list of tags

```
TagsInput::make('tags')
```

```
->suggestions([ ← add autocomplete
                suggestions
                'tailwindcss',
                'alpinejs',
                'livewire',
            ])
```

Tags

New tag

Tailwind CSS ×

Alpine.js ×

by default, tags are stored in `JSON`

Methods

```
->separator(',') store tags in a separated string, instead of JSON
->splitKeys(['Tab', ' ']) map buttons to create a new tag. By default, pressing "Enter" button, creates a new tag
->tagSuffix('%') add prefix and suffix to tags without modifying the real state of the field. Useful to show
->tagPrefix('%') presentational formatting to users without saving it
->reorderable() allow the user to reorder tags
->color('danger') change tag color. danger, gray, info, primary, success or warning
```

Validation specific to TagsInput

```
TagsInput::make('tags')
```

```
->nestedRecursiveRules([ ← add validation rules
                        for each tag by passing
                        an array of rules
                        'max:255',
                    ])
```



Add an array or json cast to the model property when using Eloquent

Utility Injection (Closure Customization)

Available utilities

Parameter		Namespace
Component \$component	current component instance	use Filament\Forms\Components\Component;
Livewire \$livewire	current Livewire component instance	use Livewire\Component as Livewire;
?Model \$record	current form record	use Illuminate\Database\Eloquent\Model;
Get \$get	retrieve the state (value) of another field	use Filament\Forms\Get;
Set \$set	set the value of another field (useful in afterStateUpdated() method)	use Filament\Forms\Set;
\$state	current state (value) of the field	-
\$old	old value of the field before it was updated (used with afterStateUpdate() method)	-
\$search	current search query (used in modifyQueryUsing on relationship())	-
\$operation	current form operation (edit, create, view)	-

in standalone form, the form's operation can be set using the `operation()` method on the form definition

Closure with utility injection

```
use Filament\Forms\Components\Component;

function (Component $component) {
    // ...
}
```

Closure with multiple utility injection

```
use Livewire\Component as Livewire;
use Filament\Forms\Get;

function (Livewire $livewire, Get $get, Set $set) {
    // ...
}
```

utility injection
(any order can be used)

Closure with Laravel's dependencies injection

```
use Illuminate\Http\Request;

function (Request $request, Set $set) {
    // ...
}
```

any dependencies from Laravel's container can also be injected

Example: form action with utility injection

```
TextInput::make('cost')
    ->prefix('€')
    ->suffixAction(
        Action::make('copyCostToPrice')
            ->icon('heroicon-m-clipboard')
            ->requiresConfirmation()
            ->action(function (Set $set, $state) {
                $set('price', $state);
            })
    )
```

utility injection